

PATENT APPLICATION

Docket No.:D428

Inventor(s): Mr. Andrew H. Quintero, Mr. Jeffrey S. Fedor,
Mr. Alan G. Quan, Ms. Karen Richardson,
Mr. Donald W. Scott, Mr. Ken A. Piper

Title: Surveillance Monitoring and Automated Reporting Method
for Detecting Data Changes

SPECIFICATION

Statement of Government Interest

The invention was made with Government support under contract
No. F04701-93-C-0094 by the Department of the Air Force. The
Government has certain rights in the invention.

Field of the Invention

The invention relates to the field of computer monitoring of
data changes. More particularly, the present invention relates to
surveillance monitoring and automated reporting of detecting
changes in monitored data well suited for reporting detected
changes in internet websites content data.

///

Background of the Invention

Electronic storage of information in computerized databases and file servers has all but replaced the traditional library as a data source of recording knowledge. Modernly, a user provides locating information about the subject matter of interest to be found in an information source. This locating information would include knowledge about the author, title, publication date, or keywords that might appear in a written abstract about the information source. The locating information describes something about the information source, and is commonly referred to as the meta data. Historically, the written word was the primary medium found in books, newspapers, magazines and other periodicals. Modernly, the types of media for recording data have expanded to include magnetic tape, photography, video tape, digital books, computer generated reports, digital audio, digital video, computerized data bases, and internet web pages. Computer based indices have replaced card catalogs as the preferred means for locating various information sources. Most of the newly recorded data is available in electronic form and available via networked computers.

Networked computers enable rapid data sharing. The network connection can be made with optical connections, copper wire connections, or can be wireless. The networks can be localized intranets referred to as local area networks. Networks can also include many external computers distributed over a wide physical

1 area as an internet, referred to as wide area networks. To share
2 data information, the networked computers use compatible
3 communications protocols. The most common protocol includes
4 hypertext transport protocol (HTTP), that uses transmission control
5 communication protocol internet protocol (TCP/IP). The largest and
6 most common collection of networked computers is the internet.
7 HTTP is the protocol that is used on the world wide web (WWW) that
8 utilizes the hypertext markup language (HTML) to format and display
9 text, audio, and video data from a data source most often using a
10 WWW browser. The most common method to display information
11 communicated through the WWW is in the form of HTML web pages.

12
13 To view web content data of a particular web page requires a
14 reference to the location of the web page. The web page content
15 data is stored electronically in memory storage devices of a web
16 server. The servers have web domain name addresses to enable
17 retrieval of the information from the local storage. If the desired
18 web content data is on the internet, the web server storing the
19 desired web content data must first be identified. On the
20 internet, computers utilize an internet protocol address (IPA)
21 unique to each web server system. Because numbers are difficult
22 for humans to remember, alias names are used in lieu of the IPA.
23 These alias names are commonly referred to as domain names. A
24 domain name service (DNS) keeps track of which IPAs are represented
25 by the respective domain names. Once a domain name is known, a
26 user can specify the exact directory path to the file of interest
27 containing the desired web content data by specifying the complete
28

1 domain name and the directories path using a uniform resource
2 locator (URLs) on the web.

3
4 To locate desired web content data at a particular URL, the
5 user would either be required to specify the exact URL and then
6 manually review the document, or perform a search based on some
7 search criteria. The most common search method employed is through
8 the use of web based search engines. Search engines typically use
9 key words in Boolean combinations to specify search criteria.
10 Boolean combined keyword searches are routinely used by users and
11 provide users with a simple and convenient way of searching for
12 desired web content data. However, Boolean combined keyword
13 searches using search engines often produce millions of URL
14 locations with many nonrelevant web pages pointing to nonrelevant
15 web content data as part of the search result. A search engine
16 match result is also referred to as hit, whether it is relevant or
17 not to the requester. A user often has to manually review many
18 nonrelevant search hits in order to locate relevant search hits.
19 Additionally, typical Boolean combined keyword searches do not
20 provide users with a convenient means to routinely search web pages
21 linked to web page hits. Human review of data is most effective at
22 determining if the source of information is appropriate for
23 required needs, but humans often lack time to perform recurring
24 searches for desired data. While a one time search may be executed
25 by a user, users often have to disadvantageously repeat the
26 identical search process, for example, on a daily basis, in order
27 to monitor changes in web content data. Web based search engines do
28 not provide a means to perform automated routine searches based

1 upon user defined search criteria. These and other disadvantages
2 are solved or reduced using the invention.

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

///

1
2 Summary of the Invention
3

4 An object of the invention is to provide a method for
5 routinely searching over a network for changes in data content.
6

7 Another object of the invention is to provide a method for
8 routinely searching data sources over a network for changes in data
9 content within defined search criteria.
10

11 Yet another object of the invention is to provide a method for
12 routine notification of changes in data content of networked data
13 sources having data content within defined search criteria.
14

15 Still another object of the invention is to provide a method
16 for routine notification of changes in data content of data sources
17 connected over a network.
18

19 A further object of the invention is to provide a method for
20 routine identification of changes in data content of networked data
21 sources identified by search criteria and having data content also
22 identified by the search criteria.
23

24 Yet a further object of the invention is to provide a method
25 for routine identification of linked data sources having data
26 content within defined search criteria.
27
28

1 Still a further object of the invention is provide a method
2 for routine notification of changes in data content of linked data
3 sources having changed data content within defined search criteria.

4
5 The invention is directed to a method for monitoring networked
6 data sources for changes in data content within defined search
7 criteria and provides users with notification of those changes. The
8 invention is applicable to both web based services and networked
9 systems for providing computer program processes that search for
10 changes in content data. The searches include conventional Boolean
11 combined keyword searches. During web based monitoring, the method
12 monitors changes data of user specified data sources that match the
13 search criteria. The data sources can be web servers identified by
14 uniform resource locators (URLs). The content data can be web
15 content data also identified by the URLs. As a stand alone process
16 executed on a networked computer of a user, the method monitors
17 other network data sources, such as other networked computers, for
18 changes in the data content of the search defined data sources.
19 For web based services, users may be given an account where the
20 users specify a list of information sources, some of which may be
21 in the form of web pages identified by the (URLs) to be monitored
22 and specify associated keywords, or other more complex criteria,
23 that are of a particular interest to the users. The method is
24 well suited for website searches. A URL is used to specify a
25 website with the URL having a http:// scheme, and having a domain
26 name for locating the website. The content data sought at the
27 website can be identified by the path extension of the URL. In the
28 general case of any networked system, a uniform resource identifier

1 with associated keywords of interest for each URL. The user
2 interface to the monitoring web server is the user web browser that
3 points to the URL of a monitoring web server. After login into the
4 monitoring web server, the user can then provide the search
5 criteria and the frequency of the searches for each specified URL
6 that is then checked for sampled for changes. The detected change
7 notification can be by way of electronic mail, pager, or a near
8 real-time graphical status display. The user can specify the
9 crawling depth of intradomain hyperlinks that will be searched for
10 occurrence of the specified keywords. The method preferably uses a
11 web server such as an apache web server that interfaces to a
12 database while executing C programs, common gateway interfaces and
13 java programs.

14
15 The method provides automatic recurring notification of search
16 result for any user that desires to stay as current as possible of
17 changing data. Web tools can be used to repetitively locate
18 networked content data with an ability to continuously monitor
19 information sources for updates, or changes, in the content data of
20 only pertinent information within the specified search criteria.
21 The method monitor changes of the web content data that are of
22 particular interest to the user on a recurring basis specified by
23 the user.

24
25 The method preferably provides a service website to the user
26 to allow the user to select URLs and corresponding keywords for
27 each URL, the crawling depth to which links will be followed for
28 keyword searching, the frequency of checking for each URL expressed

1 in minutes, hours, or days, the electronic mail, pager, or personal
2 digital assistant addresses to which notification reports will be
3 sent, the category to which the URL will be assigned, and the
4 keyword Boolean expression that will be used to search the web
5 pages. The Boolean expression allows keywords to be joined with
6 AND and OR operators. Once the URL and its parameters are defined,
7 the user then can launch or terminate the search and detection
8 process for each specified URL through the internet.

9
10 The search and detection software is implemented as a search
11 daemon that runs as an independent background process on the host
12 machine that is preferably a web server. As soon as a search
13 daemon is launched, the search daemon follows a predetermined
14 search procedure. A network connection is established to the user
15 specified URL that is to be monitored. A web request is sent over
16 the internet to download the HTML from the URL. All the characters
17 sent in response to the URL request are saved in a file. In
18 addition, a second text only file is created that contains the
19 formatted version of the text without HTML tags. To create this
20 file, while the characters are being received from the data source,
21 any text that is part of an HTML tag is not written to the text
22 only file. All other text characters are written to the file.
23 Thus, after all the HTML data is received for the URL, the text
24 only file contains all the text from the URL minus the HTML tags.
25 During the HTML acquisition, a list of all URL links that appear in
26 the web page is created for crawling through linked pages to the
27 specified crawling depth for determining if the linked pages also
28 match the specified search criteria.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Changes are detected based on a comparison of the previous text data only version of the web page stored in the database with the newly downloaded text only version of the page, both with duplicative white spaces firstly removed. The new formatted text is compared to the formatted text of the previous version for determining changes in the number of keyword hits matching the Boolean search criteria. If the current and previous text version do not match then further comparison is required in order to avoid reporting of trivial changes that the user would not be interested in. The keyword counts for the new page are determined. If any one of the keyword counts for the new page differs from the corresponding keyword count for the previous version, then a change is declared between the current and previous text only versions. After the initial comparison between the previous version in the database and the new current version is done, the previous version of the page in the database is replaced by the formatted text of the new current version. In this manner, relevant sought after changes are detected. The change detection is repeated as often as the specified search frequency. After each detection of a change in the keyword counts, the user is notified. In this manner, the monitoring method continually searches the content data for changes with automatic reporting to the user. These and other advantages will become more apparent from the following detailed description of the preferred embodiment.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Brief Description of the Drawings

Figure 1 is a block diagram of a monitored distributed network.

Figure 2 is a block diagram of a network connected monitoring and reporting system.

Figure 3 lists a top level portion of a surveillance daemon.

Figure 4A lists a pseudocode for an HTTP client data retrieval portion of a surveillance daemon subroutine.

Figure 4B lists a pseudocode for a change detection portion of the surveillance daemon subroutine.

Figure 4C lists a pseudocode for a recursion portion of the surveillance daemon subroutine.

Figure 5 lists a pseudocode for a change detection subroutine.

///

1
2 Detailed Description of the Preferred Embodiment
3

4 An embodiment of the invention is described with reference to
5 the figures using reference designations as shown in the figures.
6 Referring to Figures 1 and 2, a monitoring distributive network 10,
7 that is preferably the internet, provides interconnection between a
8 surveillance monitoring and automated reporting system 12 simply
9 also referred to as the monitoring system, and plurality of A, B,
10 and C user systems 14a, 14b, and 14c respectively, collectively
11 simply also referred to as users, and a plurality of distributed
12 networked A, B, and C monitored computer systems, 16a, 16b, 16c
13 respectively, and collectively simply also referred to as monitored
14 systems. The networked distributed computer systems 16a, 16b and
15 16c are preferably websites, but may generally be file systems,
16 databases, and/or local file systems connected to the network 10.
17 The monitored systems 16a, 16b, and 16c are monitored by the
18 monitoring system 12. The user computers 14a, 14b, and 14c connect
19 to the monitoring system 12 and the monitored systems 16a, 16b and
20 16c through the network 10. The user systems 14a, 14b, and 14c
21 respectively include an A browser 18a, a B browser 18b, and a C
22 Browser 18c, with respective data storage 20a, 20b, and 20c that
23 are typically local disk storage devices of user systems 14a, 14b,
24 and 14c.

25
26 The monitored distributed network 10 can be a network of
27 varying configurations, and can be, for example a private local
28 area network, a wide area network, or a public network, such as the

1 internet. The user systems 14a, 14b, and 14c can be workstations,
2 personal computers, or larger mainframe computer systems. Each
3 user computer 14a, 14b, and 14c typically includes one or more
4 processors, memories, and input/output devices, all well known but
5 not shown. The browsers 18a, 18b, and 18c are communication
6 interfaces to the network 10 when the monitoring system 12 is
7 particularly adapted for website communications for monitoring
8 websites that may be the monitored web server systems 16a, 16b and
9 16b, though other types of communication interfaces and information
10 systems may be used. The browser 18a, 18b, and 18c are preferably
11 particularly programmed for searching, sending and receiving web
12 content data for websites of the web servers 16a, 16b and 16c
13 located by internet protocol addresses (IPAs) on the internet. The
14 network 10 allows interconnection to a vast array of connected
15 computer systems. The monitored systems 16a, 16b, and 16c are
16 typically information storage systems but are preferably website
17 servers having respective uniform resource locators (URLs) and
18 respectively storing URL identified web content data over the world
19 wide web (WWW). The user systems 14a, 14b, and 14c access the web
20 based monitoring service of the monitoring system 12 preferably
21 using the web browsers 18a, 18b, and 18c. Although the monitoring
22 system 12 generally focuses on monitoring information systems, such
23 systems are preferably WWW website server systems. However, the
24 monitoring system 12 can also be used for monitoring information
25 through other wide or local area networks, or information stored in
26 any distal computer system using specific networking communications
27 protocols when communicating through the network 10.

28

1 Referring to all of the Figures, the monitoring system 12 is
2 preferably a website server computer system for communicating over
3 the internet when the network 10 is the internet and when the
4 monitored information systems 16a, 16b, and 16c are website
5 servers storing URL specific web content data. In the preferred
6 form, the monitoring system 12 is a web based server system
7 including a front end web server 30 for communicating over the
8 internet network 10 using URLs for defining web content data and
9 IPAs for defining website internet network address locations. The
10 monitoring system can launch and concurrently execute a plurality
11 of surveillance daemons, such as surveillance daemons 32a, 32b, and
12 32c interfacing with a database manager 34 managing a relational
13 database 36. The top level pseudocode for the surveillance daemon
14 is listed in Figure 3. Preferably, each of the surveillance daemon
15 32a, 32b and 32c concurrently communicate with a respective
16 notification daemon 38a, 38b and 38c. Each pair of surveillance
17 daemon and notification daemon respectively operates in combination
18 to respond to user monitoring requests and provide notification of
19 the monitoring results. User system 14a, 14b, and 14c, using
20 respective browser 18a, 18b, and 18c provide the monitoring system
21 12 with respective search criteria, in response to which, the
22 monitoring system 12 would invoke respective surveillance daemons
23 32a, 32b, and 32c, and respective notification daemons 38a, 38b,
24 and 38c during the monitoring process.

25
26 The monitoring system 12 preferably includes the HTTP web
27 server 30, the database manager 34, the relational database 36, and
28 one or more active surveillance daemons 32a, 32b and 32c, and one

1 or more respective notification daemons 38a, 38b and 38c, each
 2 particularly configured for web communication using URLs and IPAs
 3 over the internet network 10. The notification daemons can include
 4 sending notification of changes in web content data through
 5 electronic mail, preferably through the internet, but may also
 6 include communication through wireless devices including personal
 7 digital assistants, pagers and cell phones, and a near real-time
 8 graphical display of information source detected changes. The
 9 automated web browsers 42 of the surveillance daemons 32a, 32b, and
 10 32c, function to respectively communicate with the monitored web
 11 information systems 16a, 16b, and 16c, during searching as the
 12 change detection module 40 of the respective surveillance daemon
 13 32a, 32b and 32c function to detect change in the specified web
 14 content data. The surveillance daemon includes change detection and
 15 searching algorithms using a website monitoring code that is
 16 implemented as a software module. The notification daemons 38a,
 17 38b, and 38c function to respectively communicate with the user
 18 systems 14a, 14b, and 14c during notification of monitoring
 19 results. Each of the surveillance daemons 32a, 32b and 32c are
 20 invoked by launching the top level pseudocode of Figure 3 that can
 21 preferably launch respective surveillance daemon subroutines of the
 22 respective pseudocode listed in Figures 4A, 4B, and 4C. The
 23 surveillance daemons 32a, 32b and 32c include respective HTTP
 24 client modules 42 when executing the HTTP client portion of Figure
 25 4A of the surveillance subroutine, and have respective change
 26 detection modules 40 when executing the change detection portion of
 27 Figure 4B of the subroutine that in turn uses the recursion portion
 28 of Figure 4C and the change detection subroutine of Figure 5. The

1 HTTP client 42 can be implemented as an automated web browser. The
2 change detection module 40 and the HTTP client module 42 operate in
3 combination during monitoring with the HTTP client module fetching
4 web pages within search criteria and with the change detection
5 module determining changes in the fetched web pages.

6
7 The surveillance daemon of Figure 3 is implemented as a top
8 level pseudocode algorithm for performing basic monitoring
9 functions. Each set of user specified search criteria is
10 associated with an invoked surveillance daemon 32a 32b, or 32c at
11 line 101. Whenever the user 14a, 14b or 14c invokes a search on
12 the search criteria, a START/STOP flag in the database 36 for that
13 search criteria is set to TRUE indicating that the surveillance
14 daemon 32 has been launched for those search criteria in the
15 monitoring system 12. A RUN flag in the database 36 indicates
16 whether the surveillance daemon 32 for the search criteria is
17 currently running. When the surveillance daemon is started at line
18 100 and begins execution at line 103, the surveillance daemon first
19 sets at line 105 the RUN flag to be TRUE. The surveillance daemon
20 32 then creates a global list V at line 106 to store links that
21 have been visited during link traversal. At line 107 the
22 surveillance daemon sets a GO flag and then enters a search loop at
23 line 108 and extending to line 121 and continues to execute the
24 search loop until the surveillance daemon detects that the
25 START/STOP flag has been set to FALSE. Inside the search loop
26 between lines 108 and 121, the surveillance daemon retrieves user
27 specified information at line 110 from the database 36 specifying a
28 top level URL, a time duration between searches, and a crawling

1 depth. Next, the surveillance daemon calls at line 112 the
2 surveillance daemon SearchURL subroutine of Figures 4A, 4B and 4C,
3 with the top level URL, the crawling depth information, and the
4 current crawling level being passed as arguments to the
5 surveillance daemon SearchURL subroutine.

6
7 During surveillance daemon subroutine calls, links of the top
8 level URL are searched during link crawling and process control of
9 the subroutine terminates and process control returns to
10 surveillance daemon at line 113. At line 113, the surveillance
11 daemon checks the value of the START/STOP flag. If the START/STOP
12 flag is still TRUE at line 115, then the surveillance daemon 32
13 sleeps at line 117 for the time duration specified by the user as
14 the interval between searches. Upon waking at lines 118 and 119,
15 the surveillance daemon 32 checks the value of the START/STOP flag
16 again at line 108. If the START/STOP flag is still true at line
17 108, then the search loop starting at line 109 is executed again.
18 This search loop is repetitively executed at a frequency determined
19 by the time duration intervals that allow the surveillance daemon
20 to run continuously, checking the top level URL for changes at the
21 frequency specified by the user specified time duration. If the
22 START/STOP flag is false at line 108 when the surveillance daemon
23 awakes, then the run flag is set to FALSE at line 122 and the
24 surveillance daemon terminates execution at line 124.

25
26 The surveillance daemon 32 of top level pseudocode of Figure 3
27 calls the HTTP portion of the surveillance daemon subroutine at
28 line 112 to start execution at line 128 of the HTTP client portion.

1 At line 128, the HTTP client portion is referenced as a subroutine
2 SearchURL and begins at line 130. At line 132 a link list L is
3 created to store all HTML links that are contained in a page
4 specified by the top level URL and linked URLs. There are two
5 files that are created during the processing of the content data of
6 a top level or linked URL. A first HTML file stored in the
7 monitoring system 12 receives all of the characters that are
8 returned over the network through a network socket of the monitored
9 website specified by the top level or linked URL. The network
10 socket connection is created at line 135 to the website
11 corresponding to the top level URL or linked URL to receive the
12 HTML web content data in a buffer that forwards one character at a
13 time through a character retrieval loop of lines 139 through 157 of
14 the HTTP client portion to the HTML file stored in the monitoring
15 system 12. The entire HTML file is transferred at line 141 from
16 the buffer during a retrieval loop line 137 through line 158. A
17 second formatted text file receives the text returned from the top
18 level or linked URL with the HTML tags stripped out between lines
19 142 through 156. The formatted text (FT) file is created one
20 character at a time at lines 154 and 155. Each HTML web content
21 data character is transferred through the buffer to the HTML file
22 unconditionally at line 141. If the character is not part of an
23 HTML tag at line 142, then the character is also written to the
24 formatted text file at line 155. In order to know whether a given
25 character is within an HTML tag, a check at line 142 is done on
26 each character to see if the character marks the beginning of a
27 HTML tag. If the character marks the beginning of an HTML tag,
28 then web content data characters are read from the buffer until the

1 end of the HTML tag is found. These tag characters are written to
2 the HTML file at line 146 but not to the formatted text file. The
3 HTML tag characters are then examined at line 147 to determine if
4 the HTML tag is a link to a linked URL. If the HTML tag characters
5 are a link to a linked URL, then the linked URL is extracted from
6 the HTML tag characters and added to the end of the link list L at
7 line 149. If the HTML tag characters are not a link, then the HTML
8 tag characters form an HTML tag and are ignored. The process of
9 reading and examining HTML web content data characters is continued
10 by the loop lines 139 through 157 until all of the web content
11 characters are processed for the URL, at which time the buffer is
12 empty and the network socket is closed. The HTML file is retained
13 as a complete record in the monitoring system 12 as an exact HTML
14 copy of the web content data for the URL. The formatted text file
15 is used for all further processing by the surveillance daemon.

16
17 The formatted text file is processed in the monitoring system
18 one character at a time and stored as a single large formatted
19 string. During formatted text file processing, the formatted text
20 is formatted to eliminate excess white space at lines 160 and 161.
21 Each character that is not a white space character is appended to
22 the end of the formatted text string. Each contiguous segment of
23 white spaces in the formatted text file is converted to a single
24 blank character and then appended in order at line 160 to formatted
25 text string FS.

26
27 After creating the resulting formatted text string of the
28 pseudocode of Figure 4A, a change detection algorithm of Figure 4B

1 is called to determine if the formatted text string has changed
 2 from a previously stored formatted text string. The change
 3 detection algorithm of Figure 4B preferably only checks for change
 4 detection respecting the web content data of top level URLs at line
 5 163. If the current formatted text string is generated from a top
 6 level URL, then a change detection section of lines 166 through 183
 7 is executed. Firstly, the change detection section calls at line
 8 166 the change detection subroutine of Figure 5. The change
 9 detection subroutine of Figure 5 checks to determine if the
 10 formatted text string has changed since the last search of that top
 11 level URL, and if so, produces an updated keyword hit count and
 12 returns back to the change detection portion at line 170. The
 13 change detection portion examines the true or false result of the
 14 change detection subroutine at line 170 to determine if the change
 15 detection subroutine has determined if there has been a change
 16 since the last time that the top level URL web content data
 17 formatted text string was formatted and updated in the database 36.

18
 19 The change detection subroutine of Figure 5 returns the result
 20 of the comparison of the previous and current formatted text
 21 strings back to the calling subroutine SearchURL of Figures 4A, 4B
 22 and 4C. The flag TrueChange is set to TRUE if a significant change
 23 was detected at line 172, and if no change was detected, the flag
 24 TrueChange is set to FALSE. If a change was detected, then the new
 25 keyword counts that were generated by the change detection
 26 algorithm are added to the database, replacing the counts from the
 27 old previous version P. Then an ASCII activity report is generated
 28 at line 175. This ASCII activity report is added to the database

1 at line 176 and sent to the user at line 177 through the
2 notification method that the user has specified to be through
3 either electronic mail, pager, or personal digital assistant. When
4 a true change between the new version and previous version is
5 detected, the results are presented to the user in two different
6 formats to enable change and keyword hit notification. First, an
7 electronic message is created and sent to one or more of the user's
8 electronic mail address, pager, or personal digital assistant
9 depending on what reporting options were chosen. This message is
10 an activity report. The message should indicate that a hit has
11 occurred while specifying URLs, keywords, and the number of
12 respective keyword hits, with an abstract that includes, for
13 example, the ten words before and ten words after each keyword hit.
14 The notification may further request the user to log in to the
15 monitoring system 12 for more search result information. All
16 keyword counts should be shown. A limited number of abstracts from
17 the text may be shown as well. The abstracts may be chosen based
18 on the keywords with the highest frequency of occurrence.

19
20 The recursive portion of Figure 4C of the SearchURL subroutine
21 is executed for each of the URLs in the link list L. The change
22 detection portion jumps to line 186 when the link U1 is not the top
23 level URL, that is, when the level is greater than zero, when
24 processing each U1 link from the link list L. The change detection
25 subroutine of Figure 5 is executed once for the top level URL at
26 line 166. The top level keyword counts for the top level URL and
27 the reporting to the user between lines 170 and 184 is also
28 executed once when processing the top level URL. The processing of

1 the U1 links in list L between lines 188 and 195 and the recursive
2 portion of Figure 4C is executed for each of the U1 links in the
3 link list L. During each execution of the SearchURL subroutine for
4 each of the U1 links, the SearchURL subroutine determines the
5 number of N occurrences of each of the W keywords in each of U1
6 links of the link list L. The N occurrences of the W keywords are
7 found for each link U1 in the link list L during each recursive
8 call to the SearchURL subroutine that includes the recursive
9 portion. The change detection portion between lines 188 and 195
10 determines the N occurrences of each of the W keywords for each
11 link U1 in the link list L. The W keywords are extracted from the
12 database at line 188. The W keywords are those associated with the
13 top level URL. The N number of occurrences of each of the W
14 keywords in the U1 links are determined and added to the total
15 count T at lines 190 through 194. For each of the W keywords at
16 line 190, the N occurrences of the keyword is counted at line 192
17 to accumulate the total T keyword count for all of the W keywords
18 for each of the U1 links. The N occurrences for each of the W
19 keywords is added to the total number of keywords hits T at line
20 193. When the keyword counting is complete, T is the total number
21 of occurrences of all of the W keywords in the respective U1 link
22 being processed. The total keyword count T, the keyword occurrence
23 count N for each of the W keywords, and the crawled-to URL, that is
24 the current U1 link, are updated in the database at line 195. The
25 U1 link and the respective T total count for all of the W keywords
26 contained in the U1 link are inserted into the database for later
27 display and reporting.

28

1
2 The recursion algorithm of Figure 4C is a link traversal
3 algorithm. If flag TrueChange is TRUE at line 200, then the
4 SearchURL subroutine will attempt to traverse any links that are in
5 the page specified by the URL. All of these links are contained in
6 the previously created list L at line 149. A recursive loop at line
7 203 examines each link in list L starting at the beginning of the
8 list and first determines if the list L is empty. If the link list
9 is not empty, then the first link U1 is removed from the list at
10 line 205. A check is done at line 206 to determine if the current
11 link level is greater than or equal to the maximum crawling depth
12 for link traversal that was specified by the user. When processing
13 the top level URL, the link level is zero. If link level is less
14 than the maximum crawling depth at line 206, then the link is
15 checked to see if the link has already been processed by checking
16 if the link U1 is in the list V of visited links at line 209. If
17 link U1 is not in the list V, then the domain of U1 is determined
18 at lines 212 and 213. If the domain of link U1 matches the domain
19 of the original top level URL at line 212, then the link U1 is
20 eligible to be searched for keywords and for other links, and in so
21 doing, the link U1 will become traversed. Only links with the same
22 domain are searched in order to avoid unacceptably large link
23 search trees. The link U1 is added to list V at line 215 to show
24 that the link has been processed. A recursive call to the
25 SearchURL subroutine is performed at line 219 with arguments of
26 link U1 as the URL, crawling depth, and link level plus one because
27 the processing is progressing down one level in link traversal. The
28

1 recursion portion of the SearchURL subroutine recursively calls the
2 SearchURL subroutine for each of the URLs in the link list L.

3
4 The recursive portion of the SearchURL subroutine of Figure
5 4C, is executed at line 200 when the link level is greater than
6 zero indicating a U1 linked URL is being processed. At this point
7 the link list L contains all the links contained within the page
8 specified by URL U1. The URL, which may be the top level URL or a
9 linked URL, is examined at line 163. When the URL is a linked URL,
10 processing jumps to lines 188 through 195 to count the keywords in
11 the linked URL. During a first execution of the SearchURL
12 subroutine, when processing the top level URL, change detection is
13 performed and keywords are counted between lines 166 and 183.
14 After processing the top level URL, the recursion portion first
15 determines that there has been a true keyword change or that
16 processing is not at the top level URL of zero so that the links
17 can be processed at line 200. When the link list L is not empty,
18 and the first URL of the link list L is removed at line 205, the
19 removed U1 link is then processed. If the crawling depth of the
20 removed link has a depth less than the user specified depth at line
21 206, the removed link is compared to the domain of the top level
22 URL at lines 212 and 213. If the current depth level of the removed
23 link is less than the user specified depth, and the removed URL has
24 the same domain as the top level URL, and the URL is not in the
25 visited list V, then another recursive call to SearchURL is
26 initiated for processing the link in the link list L. This
27 recursive process continues in the loop between lines 203 to 223
28 until all the links in the link list L have been checked. During

1 each loop between lines 203 and 223, the SearchURL subroutine is
2 recursively called at line 219 to count the keywords between lines
3 188 and 195. When any link in the link list L generates a set of
4 embedded links, the embedded links are added to the link list when
5 executing the HTTP client data retrieval portion of the SearchURL
6 subroutine of Figure 4A. All of the links in the link list L are
7 processed by a recursive call of the SearchURL subroutine so that
8 the SearchURL subroutine crawls through each of the links to the
9 specified crawling depth. When the crawl level of the removed link
10 becomes equal to or greater than the specified crawling depth, then
11 the recursive call of the SearchURL subroutine will not be
12 executed. The recursive call allows link traversal to stop when
13 the SearchURL subroutine has reached the user specified crawling
14 depth. After all links in link list L have been processed, the
15 recursive call to SearchURL terminates at line 226 and control is
16 returned to line 113 of the surveillance daemon of Figure 3.

17
18 During execution of the change detection portion of the
19 SearchURL subroutine, the change detection subroutine of Figure 5
20 is called at line 166 when processing the top level URL to jump to
21 line 301 of the change detection subroutine. The change detection
22 subroutine determines true changes in the top level URL. The
23 SearchURL subroutine is repeatedly called at time intervals at line
24 112 to begin initial processing of the URL at the regular intervals
25 of sleep at line 117. During each initial processing of the top
26 level URL, the change detection portion at line 166 jumps to the
27 change detection subroutine at line 301 to begin at line 304
28 determining when there has been a true change in the top level URL.

1 During repeated monitoring of the top level URL, the text of the
2 URL may be repeatedly updated in the database. At the beginning of
3 each execution of the change detection subroutine, the previous
4 version of the text for the top level URL has been stored in the
5 database as P string. This previously stored P string is retrieved
6 at lines 306 and 307 from the database. The change detection
7 subroutine then makes direct comparison between the P string and
8 the new formatted text string FS at lines 308. If there is at least
9 one character that is different between the P string and FS string,
10 then there may be potential significant difference between the two
11 text versions that must then be processed to determine if there has
12 been a true change. The FS string replaces the P string in the
13 database at line 310 to keep the database current with the text of
14 the top level URL. To determine if there has been a true change,
15 the Boolean keyword expression (Exp) that had been previously
16 specified by the user for the top level URL is retrieved from the
17 database at lines 311 to 312. The FS string is searched at lines
18 313 for matches with Exp expression. If the expression Exp is found
19 in FS string at line 314 indicating that the W keywords exist in FS
20 in compliance with the Exp Boolean expression, then the W keywords
21 associated with the URL are retrieved from the database at line 316
22 and then, for each of the W keywords at line 317 a keyword count is
23 executed at line 319 for determining the number of occurrences of
24 each of the W keywords.

25
26 The keyword counts for the previous version P string are
27 retrieved from the database at line 321. If at least one keyword
28 count for FS is different from the corresponding keyword count for

the same keyword in the P string at line 324, then the change detection subroutine determines at line 328 that a significant difference exists between the previous P string and the new formatted FS string of the text and a true change is declared at line 328. In any other case, between lines 330 and 341, no change is declared. The change detection subroutine ends at line 344 and returns to the change detection portion where the true change is examined at line 170 and the TrueChange flag is either set to TRUE at line 172 or FALSE at line 182. In this manner, the change detection subroutine determines true changes since the last time that the top level URL was visited. After all processing for a particular top level URL is completed, including traversal of all links contained in the top level and lower level pages, the surveillance daemon then sleeps for a sleep period of time equal to the frequency interval that was specified by the user. If the user has chosen to terminate the processing of the surveillance daemon, then the surveillance daemon exits at line 124.

As may now be apparent, the surveillance daemon is used to repeatedly monitor user specified URLs at repeated user specified sleep intervals to a user specified link crawling depth searching for matches and changes in the matches to user specified keywords and keyword Boolean expressions. In the event of a change, the notification daemon provides rapid electronic notification with transmitted data so that the user can view the results. After URL monitoring notification, the user can preferably view details of the search results from a service at a website. An HTML page displaying a format similar to the electronic version can be made

1 available to the user. Preferably a page is provided to view the
 2 total keyword counts obtained from searching URL links that were
 3 followed from the top level or subsequent lower level pages during
 4 link traversal crawling. The near real time graphical status
 5 display may consist of two pop up windows that show the user two
 6 dimensional or three dimensional graphs that are repeatedly
 7 updated, for example, every sixty seconds. The graph may show the
 8 number of hits per category and the age of the data. Bars of the
 9 graph may be color coded to show aging. The combination of size
 10 and color may show the user the activity and the age of the oldest
 11 data for that category. Each bar in the graph may be clicked to
 12 bring up a new window showing either the category, one day, or one
 13 month results depending on which part of the graph is selected. A
 14 three dimensional display window may show the user the breakdown of
 15 hits and separates the hits into multiple day intervals. As may be
 16 apparent, there are many possible formats by which to display
 17 search results to the users.

1 The present invention is directed to monitoring data over a
 2 network, and preferably monitors web content data over the world
 3 wide web through internet communications using a programmed server
 4 that receives user specified search criteria including keywords,
 5 Boolean expressions, crawling depths, and sleep periods between
 6 searches, and preferably provides the user with automated
 7 notifications and website displays of the search results. The
 8 monitoring system provides the users with notification of changes
 9 in the web content data of selected websites. Those skilled in the
 10 art can make enhancements, improvements, and modifications to the
 11 invention, and these enhancements, improvements, and modifications
 12 may nonetheless fall within the spirit and scope of the following
 13 claims.